

Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/96883/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

de La Bourdonnaye, François, Setchi, Rossitza ORCID: <https://orcid.org/0000-0002-7207-6544> and Zanni-Merk, Cecilia 2016. Gaze trajectory prediction in the context of social robotics. IFAC-PapersOnLine 49 (19) , pp. 126-131. 10.1016/j.ifacol.2016.10.473 file

Publishers page: <http://dx.doi.org/10.1016/j.ifacol.2016.10.473>
<<http://dx.doi.org/10.1016/j.ifacol.2016.10.473>>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies.

See

<http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Gaze Trajectory Prediction in the Context of Social Robotics

François de La Bourdonnaye*, Rossitza Setchi**,
Cecilia Zanni-Merk****

**INSA de Strasbourg, 24 Boulevard de la Victoire, 67048 Strasbourg cedex, France*

***Cardiff University, Queen's buildings, 14-17 The Parade, Cardiff CF24 3AA, UK*

**** ICube Laboratory, 300 Boulevard Sébastien Brant, 6742 Ilkirch cedex, France*

Abstract: Social robotics is an emerging field of robotics that focuses on the interactions between robots and humans. It has attracted much interest due to concerns about an aging society and the need for assistive environments. Within this context, this paper focuses on gaze control and eye tracking as a means for robot control. It aims to improve the usability of human-machine interfaces based on gaze control by developing advanced algorithms for predicting the trajectory of the human gaze. The paper proposes two approaches to gaze-trajectory prediction: probabilistic and symbolic. Both approaches use machine learning. The probabilistic method mixes two state models representing gaze locations and directions. The symbolic method treats the gaze-trajectory prediction problem similar to how word-prediction problems are handled in web browsers. Comparative experiments prove the feasibility of both approaches and show that the probabilistic approach achieves better prediction results.

Keywords: Social robotics, human-robot interaction, eye tracking, gaze tracking, machine learning, trajectory prediction, time series, teleoperation

1. INTRODUCTION

Recent advances in domestic and humanoid robots have transformed robotics from an area primarily concerned with industrial automation to a field that supports social interactions between humans and robots. Social robotics is an emerging field that focuses on the interactions between robots and humans. It is the study of robots that interact and communicate among themselves, with humans, and with the environment, within the social and cultural structure attached to their roles (Ge and Mataric, 2009). The challenge is to overcome the existing human-robot barrier by constructing robots that behave more like humans and understand commands in an intuitive way.

The area of human-robot interaction benefits from a large range of sensors, including cameras, microphones, laser reading, and tactile sensors (Haibin, 2014). These sensors can be used for different perception tasks, including emotion recognition, object detection, face recognition, and human motion tracking. The scope of this paper is robot control using eye tracking. The use of gaze in the context of social robotics has been widely explored over the past decade. The use of gaze in teleoperation (Latif et al., 2008) and in the context of humanoid robots (Dickstein-Fischer et al., 2011) are some of the most well-known applications.

The aim of this paper is to improve existing teleoperation interfaces by developing advanced algorithms for gaze trajectory prediction. Teleoperation consists of controlling a robot where a human uses his/her gaze to draw a trajectory that is then executed by the robot. Such a scenario is based on a domestic environment where a disabled or elderly person controls a robot with his or her gaze. Gaze prediction involves communicating and understanding commands based on a

portion of a trajectory. A person does not need to complete the whole trajectory but only to start drawing its beginning for the robot to understand the entire trajectory. The approach is based on comparing real-time eye-tracking data with pre-recorded classes of gaze trajectory.

The remainder of the paper is organized as follows. Section 2 reviews the use of eye tracking in social robotics and discusses relevant approaches to gaze trajectory prediction. Section 3 highlights the research objectives and introduces the teleoperation principle as well as the concept of gaze trajectory prediction. Section 4 outlines the algorithms developed to predict gaze trajectories. Section 5 presents the experiments used to validate the prediction algorithms. The results are discussed in Section 6. The final section concludes the paper.

2. LITERATURE REVIEW

2.1 Eye Tracking in the Context of Social Robotics

The use of eye-tracking systems originates in the 1990s with the first eye-wink interfaces designed to assist severely disabled persons with their everyday activities (Shaw et al., 1990; Crisman et al., 1991). More recent examples of using eye tracking include teleoperation in surgery applications (Staub et al., 2012), navigation, and exploration (Yu et al., 2014; Latif et al., 2008, 2009a, b). In addition, eye-tracking systems are also used as therapeutic tools. For instance, Dickstein-Fischer et al. (2011) developed a humanoid robot designed to diagnose autism and interact with autism spectrum disorder (ASD) persons so that the robot could improve their social behavior. Another example is the gaze-sensitive virtual social interactive system for ASD children designed by Lahiri et al. (2011). The system uses a computer screen (instead of a humanoid robot) to interact in real time with ASD children.

More sophisticated interfaces recently developed take into account gaze direction. For instance, the gaze communication system for amyotrophic lateral sclerosis (ALS) (Maehara et al., 2003) uses a simple charge-coupled device (CCD) camera that tracks the user's gaze on a screen. In another advanced solution, the user uses his/her gaze to generate a trajectory and move a wheelchair in the real world (Antonya et al., 2011).

2.2 Gaze Prediction

Gaze prediction is a term with different meanings depending on the context of its use. It may refer to techniques used to improve eye tracking (Han, 2013); given eye-tracking data, it reduces the lag between the acquisition of the measure and its display. In this context, improving the prediction leads to improving the quality of the eye tracking.

This term can also be used in the context of egocentric videos produced by wearable cameras, such as GoPro (Li et al., 2013). The gaze-fixation prediction is computed given the wearer's head motion, hand location, and a dynamic model of the gaze.

In addition, the term is used in interactive media applications, such as bit allocation in streaming video based on region-of-interest (Feng et al., 2013). The approach uses a gaze-prediction system based on the Hidden Markov Model (HMM) where the states correspond to two of a human's intrinsic gaze behavioral movements (saccades and fixations). The parameters of the model are derived off-line from the visual saliency maps of the video. The principle of bit allocation is to allocate more bits to the regions of interest and less to other spatial regions. This is achieved so that both video compression and visual quality are improved. In exactly the same context, another gaze-location prediction application has been developed for broadcast football video (Cheng et al., 2013). The method employs a Bayesian integration of bottom-up features (motion measurements and saliency) and top-down information (ball, players, shot-type label).

In contrast to previous approaches, the approach developed in this paper considers the gaze-prediction problem to be similar to time-series prediction. Instead of using image-processing techniques, it applies machine learning on eye-tracking data to predict the human gaze trajectory.

2.3 Trajectory Prediction

The trajectory-prediction problem has been investigated from two different perspectives. First, gaze-trajectory prediction can be seen as a time-series prediction problem. A survey of methods for long-term prediction by Hellbach et al. (2009) compares several time-series prediction methods applied to prediction problems in the field of mobile robots and human-robot interaction. These include autoregressive models, local modeling, cluster-weighted modeling, and echo state networks. Experiments show that echo state networks (Yao et al., 2013) and local modeling (Oh et al., 2003) produce better results for long-term motion prediction.

Second, probabilistic methods, such as Markov Chain Models (Ishikawa et al., 2004) and Hidden-Markov Models (Qiao et

al., 2015) are widely used for trajectory prediction of moving objects and can be applied to this problem. This paper uses these methods in the development of the probabilistic algorithm.

3. CONCEPTUAL MODEL

Gaze control involves trajectory classification and trajectory prediction. This paper focuses on trajectory prediction. This section briefly describes teleoperation from the eye-data processing methods to the control of the robot. It then outlines the principle of gaze-trajectory prediction.

3.1 Teleoperation

As shown in Fig. 1, teleoperation consists of the execution by the robot of a scaled version of the trajectory drawn on a screen by a human user. The trajectory is displayed in real time on the computer screen.

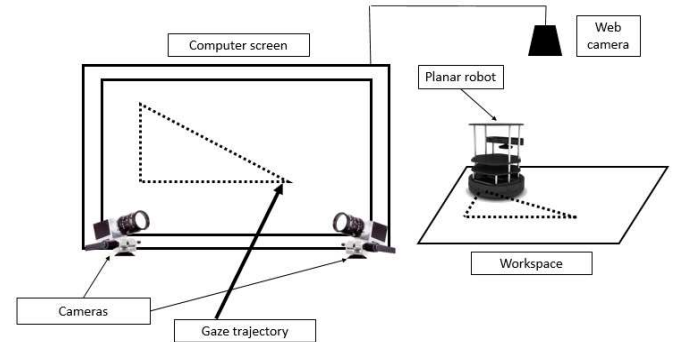


Fig. 1. Teleoperation scheme

The acquisition method is based on cascading three algorithms:

- The first algorithm filters the raw eye-tracking data. A double exponential filter is chosen because it is a good trade-off between precision and implementation simplicity. Furthermore, the fact that it is an infinite impulse response (IIR) filter makes it suitable for real-time applications.
- The second is a cartographic algorithm (Zhao and Saalfeld, 1997). This routine reduces the number of points and still conserves the topology of the trajectory. It is used to make the teleoperation more practical; indeed, the robot does not have to consider too many points.
- The third algorithm is a fixation-detection algorithm. A fixation happens when the user stares at a specific area. It produces an undesirable noise that cannot be removed by the double exponential filter. The fixation-detection algorithm detects the fixations and removes the associated noise. This paper uses a fixation-detection method based on a dispersion threshold (Salvucci and Goldberg, 2000).

The three algorithms are implemented in a cascade manner in real-time operation. Note that the method handles 3 known eye movement patterns: saccades, smooth pursuit movements, and fixations. Saccades generate rectilinear trajectories. Smooth pursuit movements imply more complex trajectories. Fixations are handled as explained before. When the trajectory is entirely

drawn, it is sent to a planar robot over the network using the Transmission Control Protocol (TCP). The robot executes the trajectory point by point using a proportional-integral-derivative (PID) controller.

3.2 Trajectory Prediction

Fig. 2 shows the principle of the gaze trajectory prediction. The scenario is very close to the teleoperation scheme in Fig.1. The only difference is that when the user draws a curve, the robot assesses which trajectory the user may want to draw. The user has the choice to continue with the trajectory or accept the prediction.

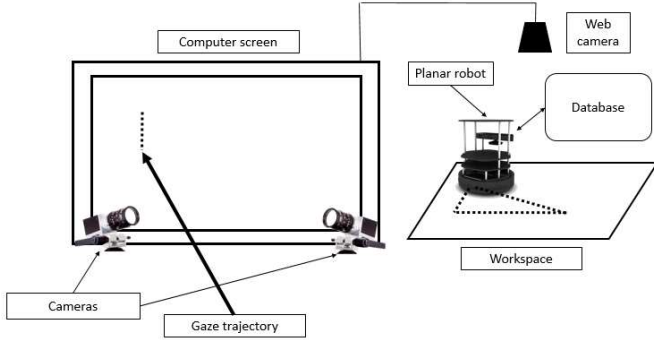


Fig. 2. Trajectory-prediction scheme

Several types or classes of trajectories are trained and recorded in a database. The trajectory is built using the inverse discrete cosine transform (IDCT) (for classification purposes, each class of trajectory is featured by DCT coefficients) and sent to the robot. The next section describes in detail the used prediction methods.

4. GAZE TRAJECTORY PREDICTION USING EYE TRACKING

4.1 Probabilistic Approach

This approach is aimed at simplifying a Markov chain model in terms of compression. The rationale is to use marginal probabilities instead of conditional probabilities in order to reduce the required storage load per class. However, the model of a class has to represent a succession of events, and this is represented through the states transition matrix in the Markov model. It is decided for this reason to mix two state models. The first model represents the locations, where the states are the possible positions in the workspace. The second model represents the directions, and the states are the directions of the possible velocity vectors. The final model represents a succession of states and sufficiently describes the trajectory both in terms of location and direction.

• Training

The training step is performed off-line. Let $P_P(i)$ and $P_D(j)$ be, respectively, the probability of the robot being in position state i and the probability of being in direction j . For a class C , the goal of the training step is to compute $P_P(C, i)$ and $P_D(C, j)$ for each position or direction state (i or j). The training is the same for the two state models. Once the gaze trajectory has been

acquired, one sweeps through the trajectory from the beginning to the end. For each model P or D and each state i or j , the probabilities are computed according to the following equations:

$$P_P(C, i) = \frac{N_i}{N}, P_D(C, j) = \frac{N_j}{N}, \quad (1)$$

where N_i is the number of times the trajectory is in state i and N is the number of points of the curve. Thus, each class of trajectory is featured by two vectors of features $V_{C,P}$ and $V_{C,D}$ of length N_P and N_D , respectively, such that

$$V_{C,P} = [P_P(C, 0) \dots P_P(C, i) \dots P_P(C, N_P - 1)], \quad (2)$$

$$V_{C,D} = [P_D(C, 0) \dots P_D(C, j) \dots P_D(C, N_D - 1)]. \quad (3)$$

• Prediction

The prediction step is performed in real time. The idea is to compute at each iteration n (each time a point is added to the trajectory) and for each class C the probability that curve u belongs to class C ($P_{u \in C}(n)$). The predicted class is the class for which $P_{u \in C}(n)$ is the maximum.

To compute $P_{u \in C}(n)$, let $P_{u \in C}(mod, n)$ be the probability that, at iteration n , trajectory u belongs to class C according to model mod (it can be the direction or the position). Then, the direction and position probabilities are combined using the geometric average:

$$P_{u \in C}(n) = \sqrt{P_{u \in C}(P, n) \cdot P_{u \in C}(D, n)}. \quad (4)$$

Computation of the probability $P_{u \in C}(mod, n)$ is processed as follows. At iteration n , if the trajectory is in state i of model mod , the intermediate probability $p_{mod}(C, n) = p(u \in i | mod, C)$, which is the probability that the robot is in a state i for class C according to model mod , is extracted from the training data. For each class C , the probability $P_{u \in C}(mod, n)$ is the geometric average of the probabilities $p_{mod}(C, n)$:

$$P_{u \in C}(mod, n) = \prod_{i=1}^n p_{mod}(C, i)^{\frac{1}{n}}. \quad (5)$$

The geometric average is used because it allows normalization of the probabilities. Indeed, if the probability is just the product of all the probabilities, it tends to zero when the trajectory length tends to infinity. Thus, the risk is that the probabilities are out of the boundaries of type “double.” Furthermore, the geometric average preserves the nature of the probability calculation; it is an increasing function and can be updated by the following recursive relation:

$$P_{u \in C}(mod, n) = P_{u \in C}(mod, n-1)^{\frac{n-1}{n}} \cdot p_{mod}(C, n)^{\frac{1}{n}}. \quad (6)$$

4.2 Symbolic Approach

The objective behind the development of this approach is twofold. On the one hand, in terms of compression, it is efficient to feature time series by symbols. A character can be encoded with one byte, whereas a double is encoded with 8 bytes. On the other hand, the problem is similar to the word-

prediction problem in web browsers, which is the original point of this approach.

- Training

The training step is also performed off-line. The idea is to assign a word to each trajectory class C . For this reason, the workspace is divided into equal area squares with one square standing for a letter. The training step is achieved by sweeping through the trajectory from the beginning to the end once the gaze trajectory has been acquired. Each trajectory is divided into equal-length segments; then, the average point of each segment is computed. The letter corresponding to a segment is the letter located the closest to the average point of the segment. As a result, each class C is featured by a word $W(C)$.

- Prediction

The prediction step is performed when the user draws a trajectory. This step is similar to the previous approach for several reasons. First, it updates the prediction scores $P_{u \in C}(i)$ for each class C at each iteration i . For this approach, the updates happen when the length of the trajectory is greater than a certain threshold. Second, the predicted class is the one having the best prediction score. Third, the geometric average is used again for the same reasons as in the previous approach. However, for this approach, the probability computation does not come from descriptive probabilities but relies on the distance between letters. These distances are recorded in an interclass distance matrix M .

The calculation of the prediction score for class C is as follows. The idea is to increasingly build a word to the curve u being drawn. Based on the trajectory location, a new letter is concatenated each time the length of u is greater than the threshold. If it is the i^{th} letter of the word, it is compared with the i^{th} letter of all the classes. For each class C , the i^{th} intermediate probability score is given by the following equation:

$$p(C, i) = \frac{D_{max} - D(C, i)}{D_{max}}, \quad (7)$$

with $D_{max} = \max(M)$ and $D(C, i)$ the distance between the i^{th} letter of class C and the i^{th} letter of the trajectory being drawn. This equation makes the intermediate probability score belonging to $[0, 1]$.

For each class C , at letter i , the prediction score is the geometric average of all the intermediate probabilities from 1 to i . This geometric average is computed by the recursive relation below:

$$P_{u \in C}(i) = P_{u \in C}(i-1)^{\frac{i-1}{i}} \cdot p_{mod}(C, i)^{\frac{1}{i}}. \quad (8)$$

5. EXPERIMENTS

The aim of the experiments is to evaluate the added value of the gaze prediction feature, and assess the performance of the two algorithms.

Smart Eye Pro software (2015) was used to acquire the raw data of the intersection between the user's gaze and the screen plane (Fig. 3). For the probabilistic approach, 15 position states and 8 direction states were used. For the symbolic approach, the workspace was divided into 28 equal-area squares. Thus, 28 different characters were used to build words. The choice of experimental parameters was made in an empirical way. The goal was to optimize the trade-off between prediction accuracy and storage load. The gaze prediction has to be efficient while ensuring a low storage load.

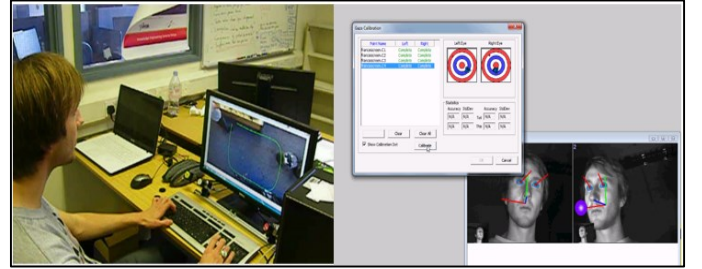


Fig. 3. The interface developed and calibration of the Smart Eye Pro software

Twelve classes of trajectory were trained in the model. The trajectories were chosen by trying to cover a large set of classes so that the experimental results could be generalized. A schematic representation of the classes is shown in Fig. 4. The black-filled dots and the arrows represent the starting points and the directions of the classes, respectively.

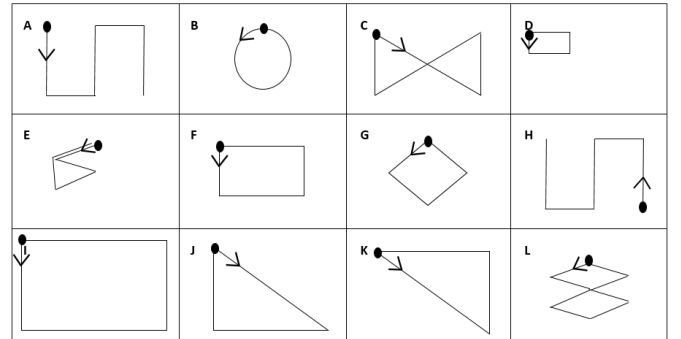


Fig. 4. Schematic representation of the classes

For each class and each approach, the prediction is checked three times. In order to check and compare the two approaches, two experimental values are used:

- The percentage of the total length of the desired class from which the desired class is among the three best-ranked classes (measure M1).
- The percentage of the total length of the desired class from which the desired class is the best-ranked class (measure M2).

These experimental values were carefully chosen to enhance the added value of the gaze trajectory prediction feature. They give an indication of how accurate and fast the prediction is. If the right class is predicted just before the end of the acquisition, the added value of the prediction feature is considered low. In contrast, if the percentage values are low,

Table 1. Average prediction of the symbolic and probabilistic algorithms

Class	Symbolic		Probabilistic	
	Among the top three classes (%)	Top class (%)	Among the top three classes (%)	Top class (%)
A	4.2	17.4	9.2	14.6
B	22.8	57.0	8.0	15.2
C	5.6	30.0	17.0	56.4
D	18.8	37.6	4.0	11.4
E	12.0	22.6	4.2	24.0
F	16.8	36.0	7.2	77.6
G	20.8	33.8	3.2	39.8
H	13.2	13.2	16.4	35.2
I	3.8	6.8	1.0	1.0
J	30.0	60.4	19.4	57.6
K	35.0	65.6	15.2	24.2
L	20.0	54.8	14.6	49.8

the long-term prediction is efficient. Two measures (M1 and M2) are chosen instead of one (M2) because the interface shows to the user the three best-ranked classes. Thus, measure M1 gives an indication of the right prediction availability speed and measure M2 evaluates more rigorously the performance of the algorithms.

6. RESULTS AND DISCUSSION

Table 2. A comparison of the two approaches

Algorithm	Maximal storage load	Quality of updating	3 rd	1 st
Symbolic	45 bytes	real time at every interval	17.17%	36.27%
Probabilistic	184 bytes	real time at every point	9.95%	33.90%

Storage load stands for the maximal storage load required per class (in bytes) to be recorded in a database. For the symbolic approach, this corresponds to the longest word (class I) among all the classes even though it is theoretically infinite. Indeed, the storage load depends on the word length; however, there is no interest in executing very long trajectories since they can be drawn in several times and it is not practical for a human user to draw long trajectories. Thus, the rectangle trajectory is assumed to be the longest possible one. For the probabilistic approach, 23 (15 position states and 8 direction states) doubles per class are needed. Thus, as double numbers are coded with 8 bytes, $23 \times 8 = 184$ bytes are required. As a consequence, in terms of storage, the symbolic approach is a better one.

In terms of updating, the probabilistic approach is better in the sense that it updates the likelihood scores for every new point, whereas the updating is achieved every equal-length interval in the symbolic approach. A user who uses the interface with the symbolic algorithm would have to wait for the likelihood updates at each space interval. This partially explains why the

Table 1 shows the average values of the percentages mentioned in the previous section for each approach (probabilistic and symbolic approaches). Column "Class" refers to the labels corresponding to the classes. The table indicates that both approaches produce satisfactory results for the 12 classes. Moreover, the results show that the probabilistic approach gives better results compared to the symbolic approach, as the percentage values are generally lower for the probabilistic method than those for the symbolic one. A comparison of the two approaches is presented in Table 2.

results are better with the probabilistic algorithm since this approach is "more reactive." The third and fourth columns correspond to the averages of the percentages mentioned above. The results from the probabilistic algorithm are better since the percentages are lower than those obtained with the symbolic approach. This means that the correct prediction is computed more quickly with the probabilistic approach. Note that these results only show that for the set of the 12 classes attempted, the probabilistic approach outperforms the symbolic one. Even if this trend is true, it has to be confirmed by further experiments on larger datasets.

7. CONCLUSION

The main contribution of this paper is the development of a long-term gaze-trajectory prediction application using eye-tracking data and machine-learning techniques. It is mainly designed for teleoperation purposes in a domestic environment. For example, a disabled or elderly person can control a robot located in another room. Two approaches have been developed. The first one, based on descriptive probabilities on kinematic models outperforms the symbolic approach. Nevertheless, the development of the symbolic approach shows that the gaze-trajectory prediction problem can be similar to the word-prediction problem in web browsers, which is itself very interesting. Furthermore, one cannot exclude any symbolic method since well-known word-prediction algorithms were not tried in this study and well-known long-term time-series prediction methods were not investigated. As a consequence, it would be interesting for future developments to compare well-known word-prediction

and long-term time-series prediction algorithms with the probabilistic approach described in this paper.

REFERENCES

- Antonya, C., Barbuceanu, F., and Rusak, Z. (2011). Path generation in virtual reality environment based on gaze analysis. *IEEE AFRICON'11*, 1-4. doi: 10.1109/AFRICON.2011.6072169
- Cheng, Q., Agrafiotis, D., Achim, A., and Bull, D. (2013). Gaze location prediction for broadcast football video, *IEEE Transactions on Image Processing*, 22(12), 4918–4929.
- Crisman, E., Loomis, A., Shaw, R., and Laszewski, Z. (1991). Using the eye wink control interface to control a powered wheelchair. *IEEE Annual Int. Conf. of the Engineering in Medicine and Biology Society*, vol.13, 1821–1822. doi:10.1109/IEMBS.1991.684772.
- Dickstein-Fischer, L., Alexander, E., Yan, X., Su, H., Harrington, K., and Fischer, G. (2011). An affordable compact humanoid robot for autism spectrum disorder interventions in children. *IEEE Annual International Conference of the Engineering in Medicine and Biology Society*, 5319–5322. doi: 10.1109/IEMBS.2011.6091316.
- Feng, Y., Cheung, G., tian Tan, W., Le Callet, P., and Ji, Y. (2013). Low-cost eye gaze prediction system for interactive networked video streaming, *IEEE Transactions on Multimedia*, 15(8), 1865–1879. doi: 10.1109/TMM.2013.2272918.
- Ge, S. and Mataric, M. (2009). Editorial to the first journal issue. *International Journal of Social Robotics*. Springer Netherlands.
- Haibin, Y., Ang Marcelo H., P.A.N. (2014) Survey on perception methods for human-robot interactions in social robots. *International Journal of Social Robotics*, 85-119.
- Han P, Saunders W.R.L.G. (2013). Trajectory prediction of saccadic eye movements using a compressed exponential model. *Journal of Vision*, 13(8).
- Hellbach, S., Eggert, J., Krner, E., and Gross, H.M. (2009). Time series analysis for long term prediction of human movement trajectories. In M. Kp- Pen, N. Kasabov, and G. Coghill (eds.), *Advances in Neuro-Information Processing, Lecture Notes in Computer Science*, vol. 5507, 567–574. Springer Berlin Heidelberg. doi:10.1007/978-3-642-03040-6.
- Ishikawa, Y., Tsukamoto, Y., and Kitagawa, H. (2004). Extracting mobility statistics from indexed spatio-temporal datasets. *2nd International Workshop on Spatio-Temporal Database Management (STDBM'04)*, Toronto, Canada, 9–16.
- Lahiri, U., Warren, Z., and Sarkar, N. (2011). Design of a gaze-sensitive virtual social interactive system for children with autism. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(4), 443–452. doi:10.1109/TNSRE.2011.2153874.
- Latif, H., Sherkat, N., and Lotfi, A. (2008). Telegaze: Teleoperation through eye gaze. *7th IEEE International Conference on Cybernetic Intelligent Systems (CIS 2008)*, 1–6. doi:10.1109/UKRICIS.2008.4798932.
- Latif, H., Sherkat, N., and Lotfi, A. (2009a). Fusion of automation and teleoperation for person-following with mobile robots. *International Conference on Information and Automation (ICIA '09)*, 1240–1245. doi: 10.1109/ICINFA.2009.5205106.
- Latif, H., Sherkat, N., and Lotfi, A. (2009b). Tele-operation through eye gaze (telegaze): A multimodal approach. *IEEE International Conference on Robotics and Biomimetics (ROBIO'09)*, 711–716. doi: 10.1109/ROBIO.2009.5420585.
- Li, Y., Fathi, A., and Reh, J. (2013). Learning to predict gaze in egocentric video. *IEEE International Conference on Computer Vision (ICCV)*, 3216–3223. doi:10.1109/ICCV.2013.399.
- Maehara, T., Uchibori, A., Mizukami, Y., Wakasa, Y., and Tanaka, K. (2003). A communication system for als patients using eye-direction. *IEEE EMBS Asian-Pacific Conference on Biomedical Engineering*, 274–275. doi:10.1109/APBME.2003.1302690.
- Oh, S.K., Seo, K.H., and Lee, J.J. (2003). Time series prediction by mixture of linear local models. *29th Annual Conference of the IEEE Industrial Electronics Society (IECON'03)*, vol. 2, 1905–1908. doi: 10.1109/IECON.2003.1280351.
- Qiao, S., Shen, D., Wang, X., Han, N., and Zhu, W. (2015). A self-adaptive parameter selection trajectory prediction approach via hidden Markov models, *IEEE Transactions on Intelligent Transportation Systems*, 16(1), 284–296.
- Salvucci, D.D. and Goldberg, J.H. (2000). Identifying fixations and saccades in eye-tracking protocols. *ACM Symposium on Eye Tracking Research & Applications, (ETRA'00)*, 71–78. doi:10.1145/355017.355028.
- Shaw, R., Crisman, E., Loomis, A., and Laszewski, Z. (1990). The eye wink control interface: using the computer to provide the severely disabled with increased flexibility and comfort. *Third Annual IEEE Symposium on Computer-Based Medical Systems*, 105–111. doi:10.1109/CBMSYS.1990.109386.
- SmartEye (2015). Source code of the SmartEye Pro software. <http://smarteyproware.informer.com/5.4>. Online; Accessed: 2015-10-25.
- Staub, C., Can, S., Jensen, B., Knoll, A., and Kohlbecher, S. (2012). Human-computer interfaces for interaction with surgical tools in robotic surgery. *4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob'12)*, 81–86. doi: 10.1109/BioRob.2012.6290850.
- Yao, W., Zeng, Z., Lian, C., and Tang, H. (2013). Ensembles of echo state networks for time series prediction, *Sixth International Conference on Advanced Computational Intelligence (ICACI)*, 299–304.
- Yu, M., Wang, X., Lin, Y., and Bai, X. (2014). Gaze tracking system for teleoperation. *26th Chinese Control and Decision Conference (CCDC)*, 4617–4622.
- Zhao, Z. and Saalfeld, A. (1997). Linear-time sleeve-fitting polyline simplification algorithms. *Autocarto*, 13, 214–223.